311-CD-506-001

EOSDIS Core System Project

Release 5A
Subscription Server Database Design and Schema Specifications for the ECS Project

Final

May 1999

Raytheon Systems Company
Upper Marlboro, Maryland

Release 5A
Subscription Server Database Design and Schema Specifications for the ECS Project

Final

May 1999

Prepared Under Contract NAS5-60000
CDRL Item #050

RESPONSIBLE ENGINEER

Maureen Muganda /s/ 5/26/99

Maureen Muganda Date
EOSDIS Core System Project

SUBMITTED BY

Mary Armstrong /s/ 5/26/99

Mary Armstrong, Systems Engineering Manager Date
EOSDIS Core System Project

Raytheon Systems Company
Upper Marlboro, Maryland

This page intentionally left blank.

Preface

This document describes the data design and database specification for the Subscription Server subsystem. It is one of eight documents comprising the detailed database design specifications for each of the ECS subsystems.

The subsystem database design specifications for the as delivered system include:

311-CD-500 Data Management (DM) Subsystem Database Design and Database Schema Specifications for the ECS Project

311-CD-501 Ingest Subsystem Database Design and Database Schema Specifications for the ECS Project

311-CD-502 Interoperability Subsystem (IOS) Database Design and Database Schema Specifications for the ECS Project

311-CD-503 Planning and Data Processing Subsystem (PDPS) Database Design and Database Schema Specifications for the ECS Project

311-CD-504 Science Data Server (SDSRV) Subsystem Database Design and Database Schema Specifications for the ECS Project

311-CD-505 Storage Management (STMGMT) Subsystem Database Design and Database Schema Specifications for the ECS Project

311-CD-506 Subscription Server (SUBSRV) Subsystem Database Design and Database Schema Specifications for the ECS Project

311-CD-507 Management Support Subsystem (MSS) Database Design and Database Schema Specifications for the ECS Project

This submittal meets the milestone specified in the Contract Data Requirements List (CDRL) of NASA Contract NAS5-60000. It is a formal contract deliverable with an approval code 2. As such, it does not require formal Government approval, however, the Government reserves the right to request changes within 45 days of the initial submittal. Once approved, contractor changes to this document are handled in accordance with Class I and Class II change control requirements described in the EOS Configuration Management Plan, and changes to this document shall be made by document change notice (DCN) or by complete revision.

Entity Relationship Diagrams (ERDs) presented in this document have been exported directly from tools and some cases contain too much detail to be easily readable within hard copy page constraints. The reader is encouraged to view these drawings on-line using the Portable Document Format (PDF) electronic copy available via the ECS Data Handling System (ECS) on the world wide web at http://edhs1.gsfc.nasa.gov.

Any questions should be addressed to:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, MD 20774-5301

Abstract

This document outlines Release 5A “as-built” database design and database schema of the Subscription Server database including the physical layout of the database and initial installation parameters.

Keywords: data, database, design, configuration, database installation, scripts, security, data model, data dictionary, replication, performance tuning, SQL server, database security, replication, database scripts

This page intentionally left blank.

Change Information Page

List of Effective Pages

Page Number	Issue
Title	Submitted as Final
iii through xii	Submitted as Final
1-1 through 1-2	Submitted as Final
2-1 through 2-2	Submitted as Final
3-1 through 3-14	Submitted as Final
4-1 through 4-2	Submitted as Final
5-1 through 5-4	Submitted as Final
6-1 through 6-2	Submitted as Final
A-1 through A-2	Submitted as Final
AB-1 through AB-2	Submitted as Final

Document History

Document Number	Status/Issue	Publication Date	CCR Number
311-CD-506-001	Submitted as Final	May 1999	99-0466

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

2. Related Documents

3. Data Design

4. Performance and Tuning Factors

5. Database Security

5. Scripts

List of Figures

List of Tables

Appendix A. Subscription Server Subsystem Entity Relationship Diagrams

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Identification

This Subscription Server (SUBSRV) Database Design and Database Schema Specification document, Contract Data Requirement List (CDRL) Item Number 050, whose requirements are specified in Data Item Description DID 311/DV2, is a required deliverable under the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS), Contract NAS5-60000.

1.2 Scope

The SUBSRV Database Design and Database Schema Specification document describes the data design and database specifications to support the data requirements of Release 5 SUBSRV software.

1.3 Purpose

The purpose of the SUBSRV Database Design and Database Schema Specification document is to support the maintenance of SUBSRV data and databases throughout the life cycle of ECS. This document communicates the database implementation in sufficient detail to support ongoing configuration management.

1.4 Audience

This document is intended to be used by ECS maintenance and operations staff. The document is organized as follows:

Section 1 provides information regarding the identification, purpose, scope and audience of this document.

Section 2 provides a listing of the related documents, which were used as a source of information for this document.

Section 3 contains the SUBSRV physical data model which is the database tables, triggers, stored procedures, and flat files.

Section 4 provides a description of database performance and tuning features such as indexes, caches, and data segments.

Section 5 provides a description of the security infrastructure used and a list of the users, groups, and permissions available upon initial installation.

Section 6 provides a description of database and database related scripts.

This page intentionally left blank.

2. Related Documents

2.1 Applicable Documents

The following documents, including Internet links, are referenced in the SUBSRV Database Design and Database Schema Specification, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume. Internet links cannot be guaranteed for accuracy or currency.

920-TDG-009	GSFC Release B0 DAAC Database Information
920-TDN- 009	NSIDC Release B0 DAAC Database Information
920-TDE-009	EDC Release B0 DAAC Database Information
920-TDL-009	LARC Release B0 DAAC Database Information
920-TDS-009	SMC Release B0 DAAC Database Information
920-TDM-009	Mini-DAAC Release B0 Database Information
920-TDG-001	GSFC Version 2.0 Hardware Diagram
920-TDN-001	NSIDC Version 2.0 Hardware Diagram
920-TDE-001	EDCC Version 2.0 Hardware Diagram
920-TDL-001	LARC Version 2.0 Hardware Diagram
920-TDS-001	SMC Version 2.0 Hardware Diagram
920-TDM-001	Mini-DAAC Version 2.0 Hardware Diagram
920-TDG-002	GSFC Version 2.0 Hardware Software Mapping
920-TDN-002	NSIDC Version 2.0 Hardware Software Mapping
920-TDE-002	EDC Version 2.0 Hardware Software Mapping
920-TDL-002	LARC Version 2.0 Hardware Software Mapping
920-TDS-002	SMC Version 2.0 Hardware Software Mapping
920-TDM-002	Mini-DAAC Version 2.0 Hardware Software Mapping

2.2 Information Documents

The following documents, although not directly applicable, amplify or clarify the information presented in this document. These documents are not binding on this document.

311-CD-500	Data Management (DM) Subsystem Database Design and Database Schema Specifications for the ECS Project
311-CD-501	Ingest Subsystem (INS) Database Design and Database Schema Specifications for the ECS Project
311-CD-502	Interoperability Subsystem (IOS) Database Design and Database Schema Specifications for the ECS Project
311-CD-503	Planning and Data Processing Subsystem (PDPS) Database Design and Database Schema Specifications for the ECS Project
311-CD-504	Science Data Server (SDSRV) Subsystem Database Design and Database Schema Specifications for the ECS Project
311-CD-505	Storage Management (STMGT) Subsystem Database Design and Database Schema Specifications for the ECS Project
311-CD-507	Management Support Subsystem (MSS) Database Design and Database Schema Specifications for the ECS Project

3. Data Design

3.1 Database Overview

Data requirements for SUBSRV span two logical grouping areas:

Event information – data pertaining to defined events

Subscription information – data pertaining to subscriptions

Database versioning information -

The SUBSRV database implements the large majority of the persistent data requirements for the SUBSRV subsystem. The database is designed in such a manner as to satisfy business policy while maintaining data integrity and consistency. Database tables are implemented using the Sybase Relational Database Management system (RDBMS). All components of the SUBSRV database are described in the sections which follow.

3.1.1 Physical Data Model Entity Relationship Diagram

The Entity Relationship Diagram (ERD) presents a schematic depiction of the SUBSRV physical data model. The ERDs presented here for the SUBSRV database were produced using the S-Designor Data Architect Computer Aided Software Engineering (CASE) tool. ERDs represent the relationship between entities or database tables. The key for the symbols used in the ERDs follows.

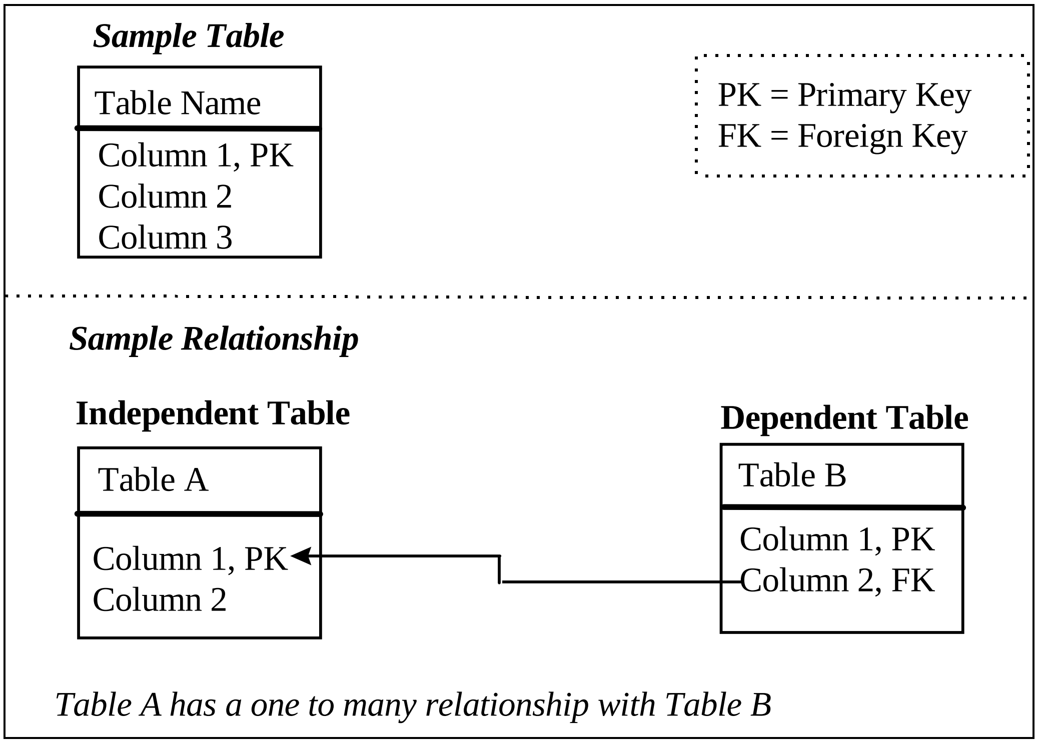


Figure 3-1. ERD Key

The ERDs for the SUBSRV database are shown in Appendix A.

3.1.2 Tables

A listing of each of the tables in the SUBSRV database is given in table 3-1. A brief definition of each of these tables follows.

Table 3-1. SUBSRV Database Tables Listing

Table Name	Logical Grouping
EcDbVersions	Database versioning
EcSbEvent	Event Information
EcSbNewEventID	Event Information
EcSbNewSubID	Subscription Information
EcSbSubscription	Subscription Information

Table 3-2. EcDbVersions

Description: This identifies the current version level of the SUBSRV database

Column List

Name	Type	PK	Mandatory
EcDbSchemaVersionID	Smallint	Yes	Yes
EcDbDropVersion	Char(64)	No	Yes
EcDbDropDescription	Varchar(255)	No	Yes
EcDbCurrentVersionFlag	Char(1)	No	Yes
EcDbDatabaseName	Varchar(255)	No	No
EcDbDropInstallDate	Datetime	No	No
EcDbSybaseVersion	Varchar(255)	No	No
EcDbSybaseServer	Varchar(255)	No	No
EcDbComments	Varchar(255)	No	No
EcDbUpdateProcess	Varchar(255)	No	No

Table 3-3. EcSbEvent

Description

Contains the list of events to which a user, or another subsystem can subscribe.

Column List

Column	Type	PK	Mandatory
Category	varchar(35)	No	Yes
EventID	int	Yes	Yes
Object	text	No	Yes
UserID	varchar(12)	No	Yes

Table 3-4. EcSbNewEventID

Description

This table is used to generate the next available ID for the EcSbEvent table.

Column List

Column	Type	PK	Mandatory
ID	Int	No	No

Table 3-5. EcSbNewSubD

Description

This table is used to generate the next available ID for the EcSbSubscription table.

Column List

Column	Type	PK	Mandatory
ID	int	No	No

Table 3-6. EcSbSubscription

Description

This table lists all the user and subsystem subscriptions. Each event can have many subscriptions. Each user can have many subscriptions. The same user can subscribe to the same event with different constraints. It is also possible that a user could subscribe to the same event with the same constraints.

Column List

Column	Type	PK	Mandatory
EventID	int	No	Yes
ExpDate	datetime	No	Yes
Object	text	No	Yes
SubID	int	Yes	Yes
UserID	varchar(30)	No	Yes

3.1.3 Columns

Brief definitions of each of the columns present in the database tables defined above are contained herein.

Column: Category

Description: Qualifier describing the type or category of a subscribable event. Currently not used.

Column: EcDbComment

Description: Notes or comments on the database version level.

Column: EcDbCurrentVersionFlag

Description: Flag indicating if this row represents the current database version entry.

Valid Values: 1= yes, 0 = no

Column: EcDbDatabaseName

Description: The name of the database for which this database version level is applied.

Column: EcDbDropDescription

Description: The official description of the ECS software drop for this database version level.

Column: EcDbDropInstallDate

Description: The date and time that the database version level was installed.

Column: EcDbDropVersion

Description: The official name of the ECS software drop for this database version level.

Column: EcDbSchemaVersionId

Description: The subsystem-specific identifier for this database schema version.

Column: EcDbSybaseServer

Description: The name of the baseline Sybase SQL server controlling this database.

Valid Values: See 920-TDx-009

Column: EcDbSybaseVersion

Description: The software release version of the Sybase SQL server in place when this database version level was initially installed.

Column: EcDbUpdateProcess

Description: The installation method by which this database version level was installed

Column: EventID

Description: Unique identifier of the event.

Column: expDate

Description: Date and time that the subscriptions expire. Default is today. Must be >= today.

Column: ID

Description: The identification number available for the next subscription generated.

Column: object

Description: Event information including qualifiable metadata.

Column: subID

Description: Unique identifier of the subscription.

Column: UserID

Description: User registering the event.

3.1.4 Column Domains

Domains specify the ranges of values allowed for a given table column. Sybase supports the definition of specific domains to further limit the format of data for a given column. Sybase domains are, in effect, user-defined data types. There are no domains defined in the SUBSRV database.

3.1.5 Rules

Sybase supports the definitions of rules. Rules provide a means for enforcing domain constraints on a given column. There are no rules defined in Sybase for the SUBSRV database.

3.1.6 Defaults

Defaults are used to supply a value for a column when one is not defined at insert time. There are no defaults defined in Sybase in the SUBSRV database.

3.1.7 Views

Sybase allows the definition of views as a means of limiting an application or users access to data in a table or tables. Views create a logical table from columns found in one or more tables. There are no views defined in the SUBSRV database.

3.1.8 Integrity Constraints

Sybase allows the enforcement of referential integrity via the use of declarative integrity constraints. Integrity constraints allow the SQL server to enforce primary and foreign key integrity checks without automatically without requiring programming. constraints support “restrict-only” operations. This means that a row can not be deleted or updated if their are rows in other tables having a foreign key dependency on that row. Cascade delete and update operations can not be performed if a declarative constraint has been used. There are no declarative integrity constraints defined in the SUBSRV database.

3.1.9 Triggers

Sybase supports the enforcement of business policy via the use of triggers. A trigger is best defined as set of activities or checks that should be performed automatically when ever a row is inserted, updated, or deleted from a given table. Sybase allows the definition of insert, update, and delete trigger per table. No triggers are currently defined in the SUBSRV database.

3.1.10 Stored Procedures

Sybase also includes support for business policy via the use of stored procedures. Stored procedures are typically used to capture a set of activities or checks that will be performed on the database repeatedly to enforce business policy and maintain data integrity. Stored procedures are parsed and compiled SQL code that reside in the database and may be called by name by an application, trigger or another stored procedure. A listing of each of the stored procedures in the SUBSRV database is given in Table 3-7. A brief definition of each of these stored procedures follows.

Table 3-7. Stored Procedure Listing

Name	Description
ProcGetAllEvents	Retrieves all registered events.
ProcGetAllSubs	Retrieves all existing subscriptions.
ProcGetCatEvents	Retrieves all events for a given category.
ProcGetEvent	Retrieves a specific event.
ProcGetEventID	Returns the next available event ID.
ProcGetEventIDSubs	Selects subscriptions made against a specific event.
ProcGetExpSubs	Retrieves events scheduled to expire on a specific date.
ProcGetSub	Retrieves a specific subscription.
ProcGetSubID	Returns the next available subscription ID.
ProcGetUIDEvents	Retrieves events for a specific user.
ProcGetUserIDSubs	Retrieves subscriptions for a specific user.
ProcRemoveEvent	Deletes a specific event.
ProcRemoveSub	Deletes a specific subscription.

3.1.10.1 Procedure: ProcGetAllEvents

Code

```
create proc ProcGetAllEvents
as
  select Object from EcSbEvent
  if @@error != 0
  begin
   raiserror 110022 "ProcGetAllEvents: can not access EcSbEvent table "
   return -999
  end
return
go
```

3.1.10.2 Procedure: ProcGetAllSubs

Code

```
create proc ProcGetAllSubs
as
  select object from EcSbSubscription
  if @@error != 0
  begin
   raiserror 110026 "ProcGetAllSubs: can not access EcSbSubscription table "
   return -999
  end
return
go
```

3.1.10.3 Procedure: ProcGetCatEvents

Code

```
/* This Proc will be deleted as the new schema is updated */
create proc ProcGetCatEvents (@Category varchar(35))
as
  select Object from EcSbEvent
  where Category = @Category
return
go
```

3.1.10.4 Procedure: ProcGetEvent

Code

```
create proc ProcGetEvent (@EventID int)
as
  if (@EventID <= 0)
   begin
    raiserror 110024 "Procedure ProcGetEvent expects parameter @EventID, which was not
supplied."
    return -900
   end
  select Object from EcSbEvent
  where EventID = @EventID
  if @@error !=0
    begin
     raiserror 110022 "ProcGetEvent: can not access EcSbEvent table "
     return -999
    end
return
go
```

3.1.10.5 Procedure: ProcGetEventID

Code

```
create proc ProcGetEventID
as
            begin transaction pubs2
            update EcSbNewEventID set ID = ID + 1
            select ID from EcSbNewEventID
commit pubs2
return
go
```

3.1.10.6 Procedure: ProcGetEventIDSubs

Code

```
create proc ProcGetEventIDSubs(@eventID int)
as
  if (@eventID <= 0)
   begin
    raiserror 110028 "Procedure ProcGetEventIDSubs expects parameter @eventID, which was
not supplied. "
    return -900
   end
  select object from EcSbSubscription
  where eventID = (@eventID)
  if @@error != 0
    begin
     raiserror 110122 "ProcGetEventIDSubs: can not access EcSbSubscription table "
     return -999
    end
return
go
```

3.1.10.7 Procedure: ProcGetExpSubs

Code

```
create proc ProcGetExpSubs (@expDate   datetime)
as
  if (@expDate is null)
   begin
    raiserror 110027 "Procedure ProcGetExpSubs expects parameter @expDate, which was not
supplied. "
    return -900
   end
  select object from EcSbSubscription
  where expDate = (@expDate)
  if @@error != 0
    begin
     raiserror 110122 "ProcGetExpSubs: can not access EcSbSubscription table "
     return -999
    end
return
go
```

3.1.10.8 Procedure:ProcGetSub

Code

```
create proc ProcGetSub(@subID int)
as
  if (@subID <= 0)
   begin
    raiserror 110030 "Procedure ProcGetSub expects parameter @subID, which was not supplied.
"
    return -900
   end
  select object from EcSbSubscription
  where subID = (@subID)
  if @@error != 0
    begin
     raiserror 110122 "ProcGetSub : can not access EcSbSubscription table "
     return -999
    end
return
go
```

3.1.10.9 Procedure: ProcGetSubID

Code

```
create proc ProcGetSubID
as
            begin transaction pubs2
            update EcSbNewSubID set ID = ID + 1
            select ID from EcSbNewSubID
commit pubs2
return
go
```

3.1.10.10 Procedure: ProcGetUIDEvents

Code

```
create proc ProcGetUIDEvents (@UserID  varchar(12))
as
  select Object from EcSbEvent
  where UserID = @UserID
return
go create proc ProcGetUIDEvents (@UserID  varchar(12))
as
  if (@UserID is null)
```

```
   begin
    raiserror 110023 "Procedure ProcGetUIDEvents expects parameter @UserID, which was not
supplied."
    return -900
   end
  select Object from EcSbEvent
  where UserID = @UserID
  if @@error !=0
    begin
     raiserror 110022 "ProcGetUIDEvents: can not access EcSbEvent table "
     return -999
    end
return
go
```

3.1.10.11 Procedure: ProcGetUserIDSubs

Code

```
create proc ProcGetUserIDSubs(@userID varchar(30))
as
  if (@userID is null)
   begin
    raiserror 110029 "Procedure ProcGetUserIDSubs expects parameter @userID, which was not
supplied. "
    return -900
   end
  select object from EcSbSubscription
  where userID = (@userID)
  if @@error != 0
    begin
     raiserror 110122 "ProcGetUserIDSubs: can not access EcSbSubscription table "
     return -999
    end
return
go
```

3.1.10.12 Procedure: ProcRemoveEvent

Code

```
create proc ProcRemoveEvent (@EventID  int)
as
if (@EventID <= 0)
    begin
     raiserror 110025 "Procedure ProcRemoveEvent expects parameter @EventID, which was not supplied."
     return -900
    end
  delete EcSbEvent
  where EventID = (@EventID)
  if @@error != 0
    begin
     raiserror 110022 "ProcRemoveEvent: can not access EcSbEvent table "
     return -999
    end
return
go
```

3.1.10.13 Procedure: ProcRemoveSub

Code

```
create proc ProcRemoveSub (@subID int)
as
  if (@subID <= 0)
   begin
    raiserror 110031 "Procedure ProcRemoveSub expects parameter @subID, which was not supplied. "
    return -900
   end
  delete EcSbSubscription
  where subID= (@subID)
  if @@error != 0
    begin
     raiserror 110122 "ProcRemoveSub: can not access EcSbSubscription table "
     return -999
    end
return
go
```

3.2 File Usage

There are cases when the implementation of a persistent data requirement is better suited to a flat file than to a database table. A typical example of such data is system configuration information. System configuration information is fairly static and usually has no explicit relationship to other data in the enterprise. Another common use of files in ECS is as an interface mechanism between ECS and the external world. There are no flat files used in SUBSRV.

3.2.1 Files Definitions

Not applicable.

3.2.2 Attributes

Not applicable.

3.2.3 Attribute Domains

Not applicable.

4. Performance and Tuning Factors

4.1 Indexes

An index provides a means of locating a row in a database table based on the value of a specific column(s), without having to scan all data in the table. When properly implemented, indexes can significantly decrease the time it takes to retrieve data, thereby increasing performance. Sybase allows the definition of two types of indexes, clustered and non-clustered.

In a clustered index, the rows in a database table are physically stored in sequence-determined by the index. Clustered indexes are particularly useful, when the data is frequently retrieved in sequential order. Only one clustered index may be defined per table.

Non-clustered indexes differ from their clustered counterpart, in that, data is not physically stored in sorted order—newly added rows are stored at the end of the related database table.

A key of the types of indexes found in SUBSRV is provided in Table 4-1 Index Type Key. A description of each of the defined indexes is given in Table 4-2 Index List

Table 4-1. Index Type Key

Index Type Key	Description
PK	Primary Key
FK	Foreign Key
U	Unique - Only one for the column code combination
C	Clustered or non-clustered index
Sort	ASC (ascending) of DESC (descending) order

Table 4-2. Index Listing

Table Code	Index Code	Primary Key	Foreign Key	Unique	Clustered
EcSbEvent	IndexEventID IndexUID IndexCategory	Yes No No	No No No	Yes No No	No Yes No
EcSbNewEventID					
EcSbNewSubID					
EcSbSubscription	EcSbSubscr_160030881 eventIDIndex expDateIndex userIDIndex	Yes No No No	No No No No	Yes No No No	Yes No No No

4.2 Segments

Sybase supports the declaration of segments. A segment is a named pointer to a storage device(s). Segments are used to physically allocate a database object to a particular storage device. Segments defined for the SUBSRV and all other subsystem databases are described in Table 4-2.

Table 4-3. Segment Descriptions

Segment Name	Description
default	Default data segment used if no other segment specified in the create statement.
logsegment	SYSLOGS, Transaction Logs
systemsegment	System tables and indexes.
SUBOPSDAT01	SUBSRV OPS mode data segment.
SUBOPSIDX01	SUBSRV OPS mode index segment.
SUBTS1DAT01	SUBSRV TS1 mode data segment.
SUBTS1IDX01	SUBSRV TS1 mode index segment.
SUBTS2DAT01	SUBSRV TS2 mode data segment.
SUBTS2IDX01	SUBSRV TS2 mode index segment.

4.3 Caches

A cache is a block of memory that is used by Sybase to retain and manage pages that are currently being processed. By default, each database contains three caches:

Data cache – retains most recently accessed data and index pages

Procedure cache – retains most recently accessed stored procedure pages

User transaction log cache – transaction log pages that have not yet been written to disk for each user

The size of each of these default caches is a configurable item which must be managed on a per DAAC basis. These caches may be increased or decreased by the DAAC DBA as needed.

The data cache can be further subdivided into named caches. A *named cache* is a block of memory that is named and used by the DBMS to store data pages for select tables and/or indexes. Assigning a database table to named cache causes accessed pages to be loaded into memory and retained. The named cache does not need to be allocated to accommodate the entire database table since the DBMS manages the cache according to use. Named caches greatly increase performance by eliminating the time associated for disk input and output (I/O). There are no named caches that are currently defined for the SUBSRV Subsystem database. Named caches may be defined as the memory usage of the SUBSRV database becomes more well known and the DAACs move into an operational environment. As named caches are defined this portion of the document will be updated.

There are no named caches for the subscription server database.

5. Database Security

5.1 Approach

The database security discussed within this section is bounded to security implementation within the Sybase SQL Server DBMS. A Sybase general approach to security is adopted as illustrated in Figure 5-1.

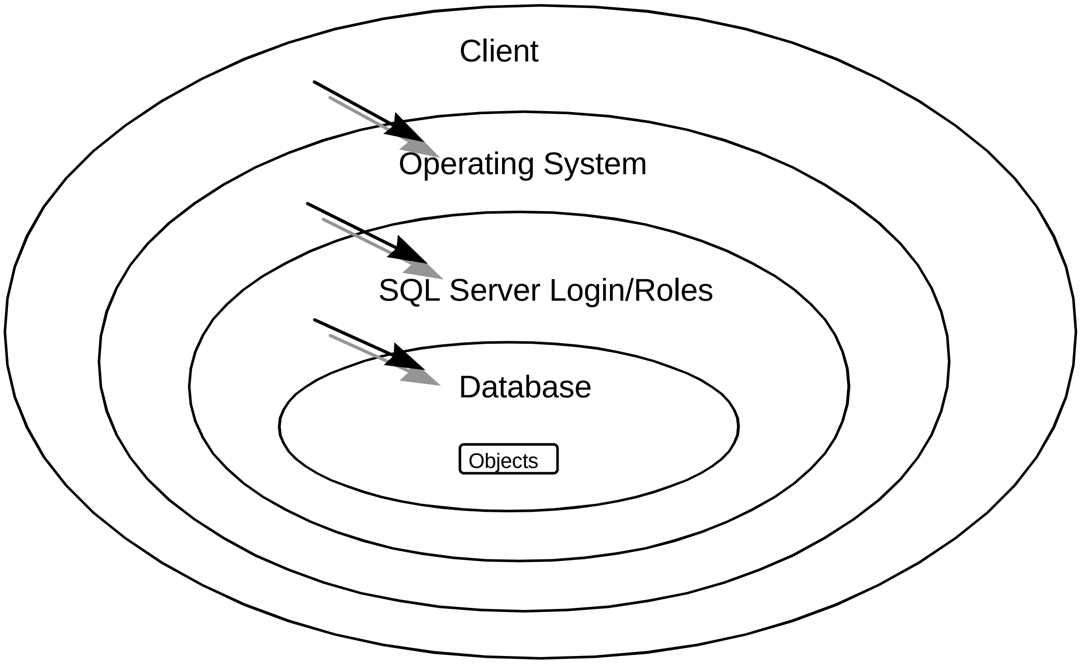


Figure 5-1. Sybase General Approach to SQL Server Security[1]

5.2 Users

The client (user) requires a SQL Server login to access the DBMS The login is assigned to a user with certain related permissions for gaining access to particular objects (e.g., database tables, views, commands) within the database. The System Administrator may grant or revoke objects permissions for a login individually or based on defined group or roles.

[1] Reference Sybase Student Guide: *Advanced SQL Server A<SUBSYS>inistration*.

5.3 Groups

Groups are a means of logically associating users with similar data access needs. Once a group has been defined, object and command permissions can be granted to that group. A user who is member of a group inherits all of the permissions granted to that group. No groups have been initially defined in the SUBSRV Subsystem "default database. The DAACs should define database groups to support the database security requirements of their individual DAACs. Security for local DAAC users should be controlled by assigning each user to the appropriate group.

5.4 Roles

Roles were introduced in Sybase to allow a structured means for granting users the permissions needed to perform standard database administration activities and also provide a means for easily identifying such users. There are six pre-defined roles that may be assigned to a user. A definition of each of these roles follows, as well as a description of the types of activities that may be performed by each role.

System Administrator (*sa_role*): This role is used to grant a specific user permissions needed to perform standard system administrator duties including:

- installing SQL server and specific SQL server modules
- managing the allocation of physical storage
- tuning configuration parameters
- creating databases

Site Security Officer (*sso_role*): This role is used to grant a specific user the permissions needed to maintain SQL server security including:

- adding server logins
- administrating passwords
- managing the audit system
- granting users all roles except the *sa_role*

Operator (*oper_role*): This role is used to grant a specific user the permissions needed to perform standard functions for the database including:

- dumping transactions and databases
- loading transactions and databases

Navigator (*navigator_role*): This role is used to grant a specific user the permissions needed to manage the navigation server.

Replication (*replication_role*): This role is used to grant a specific user the permissions needed to manage the replication server.

Sybase Technical Support (*sybase_ts_role*): This role is used to grant a specific user the permissions needed to execute *database consistency checker (dbcc)*, a Sybase supplied utility supporting commands that are normally outside of the realm of routine system administrator activities.

The DAACs should review these roles and assign them to the appropriate login and/or groups.

5.5 Login/Group Object Permissions

During initial database installation logins used by the ECS custom code were created and permissions assigned for access to the SUBSRV Subsystem database. In addition, special database installation login, subsrv_role, was created to support database installation needs. For each login, the level of access is limited to that associated with their login, group or assigned group/role. Object Permissions are set within the installation scripts of the SUBSRV Subsystem for each object and group/role.

Permissions are identified in Table 5-1. A specification of the object permissions is contained in Table 5-2.

Table 5-1. Permission Key

Permission	Description
A	All
S	Select
I	Insert
U	Update
D	Delete
E	Execute

Table 5-2. Object Permissions (1 of 2)

Group/User	Sybase Login	Object	Grant				
			Select	Insert	Update	Delete	Execute
public		ProcGetAllEvents					G
public		ProcGetAllSubs					G
public		ProcGetCatEvents					G
public		ProcGetEvent					G
public		ProcGetEventID					G

Table 5-2. Object Permissions (2 of 2)

Group/User	Sybase Login	Object	Grant				
public		ProcGetEventIDSubs					G
public		ProcGetExpSubs					G
public		ProcGetSub					G
public		ProcGetSubID					G
public		ProcGetUIDEvents					G
public		ProcGetUserIDSubs					G
public		ProcRemoveEvent					G
public		ProcRemoveSub					G
public		Table:					
public		EcSbEvent	G	G	G	G	
public		EcSbNewEventID	G	G	G	G	
public		EcSbNewSubID	G	G	G	G	
public		EcSbSubscription	G	G	G	G	
sa_role	subsrv_role	all					

6. Scripts

6.1 Installation Scripts

Any scripts used to support installation of the SUBSRV database are described herein. These files are found in the directory /ecs/formal/CSS/DOF/src/SUBSCRIPTION/sybase.

Script File	Description
EcCsSbDbBuild	Installs/populates Subscription Server database

6.2 De-Installation Scripts

Any scripts used to support de-installation of the SUBSRV database are described herein.

Script File	Description
EcCsSbDbDrop	Drops database objects

6.3 Backup/Recovery Scripts

Any scripts used to facilitate backup or recovery of the SUBSRV database are described herein.

Script File	Description
EcCsSbDbDump	Creates a backup of the database
EcCsSbDbLoad	Restores the database

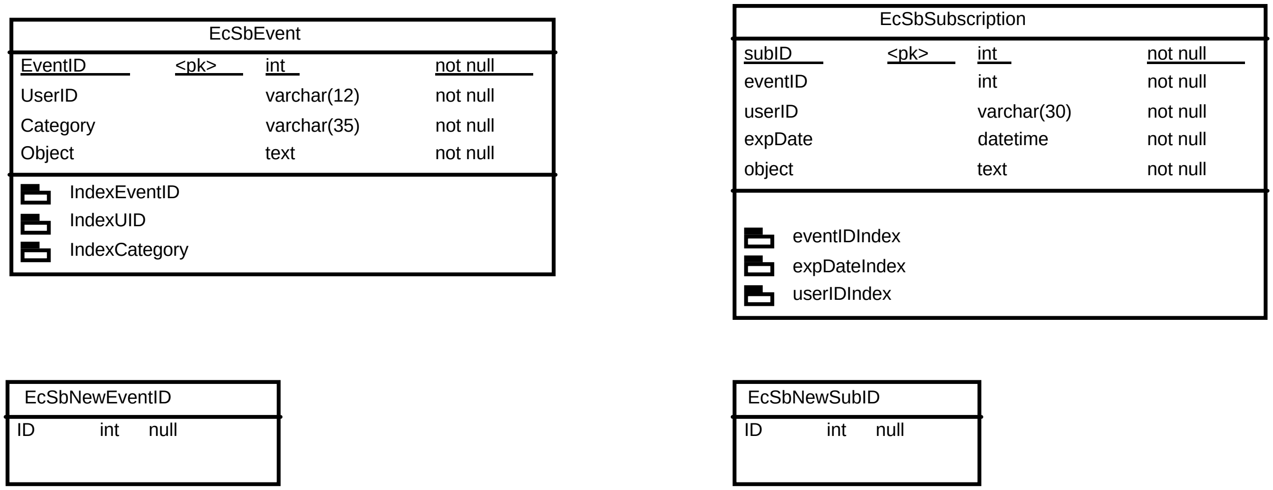
6.4 Miscellaneous Scripts

Miscellaneous scripts applicable to the SUBSRV database are described herein.

Script File	Description
EcCsSbDbPatch	Install database schema modifications

This page intentionally left blank.

Appendix A. Subscription Server Subsystem Entity• Relationship Diagrams•



EcSbEvent			
EventID	<pk>	int	not null
UserID		varchar(12)	not null
Category		varchar(35)	not null
Object		text	not null

IndexEventID
IndexUID
IndexCategory

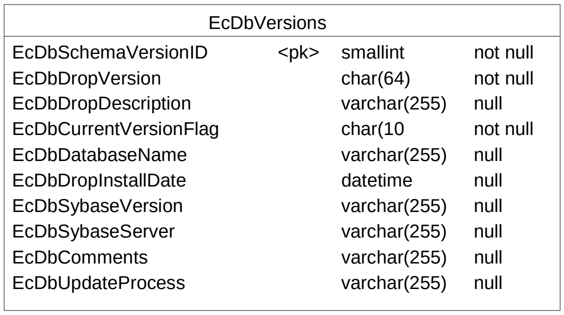
EcSbSubscription			
subID	<pk>	int	not null
eventID		int	not null
userID		varchar(30)	not null
expDate		datetime	not null
object		text	not null

eventIDIndex
expDateIndex
userIDIndex

EcSbNewEventID		
ID	int	null

EcSbNewSubID		
ID	int	null

Figure A-1. Subscription Server



EcDbVersions			
EcDbSchemaVersionID	<pk>	smallint	not null
EcDbDropVersion		char(64)	not null
EcDbDropDescription		varchar(255)	null
EcDbCurrentVersionFlag		char(10	not null
EcDbDatabaseName		varchar(255)	null
EcDbDropInstallDate		datetime	null
EcDbSybaseVersion		varchar(255)	null
EcDbSybaseServer		varchar(255)	null
EcDbComments		varchar(255)	null
EcDbUpdateProcess		varchar(255)	null

Figure A-2. Database Versions

This page intentionally left blank.

Abbreviations and Acronyms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Exchange
CASE	Computer Aided Software Engineering
CD	contractual delivery 213-001
CDRL	contract data requirements list
CI	configuration item
COTS	commercial off-the-shelf (hardware or software)
CSCI	computer software configuration item
DAAC	Distributed Active Archive Center
DBCC	Database Consistency Checker
DBMS	Database Management System
DCN	Document Change Notice
DID	data item description
DMS	Data Management Subsystem
ECS	EOSDIS Core System
EDC	EROS Data Center
EDHS	ECS Data Handling System
EOSDIS	Earth Observing System Data and Information System
EROS	Earth Resources Observation System
ERD	Entity Relationship Diagram
ESDIS	Earth Science Data and Information System (GSFC)
ESDT	Earth science data types
ESN	EOSDIS Science Network (ECS)
FK	Foreign Key
GSFC	Goddard Space Flight Center
GUI	graphic user interface
HDF	hierarchical data format

HDF-EOS	an EOS proposed standard for a specialized HDF data format
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
I/O	input/output
ICD	interface control document
INGST	Ingest Services CSCI
IOS	Interoperability Subsystem
LaRC	Langley Research Center (DAAC)
MSS	Management Support Subsystem
N/A	not applicable
NAS	National Academy of Science
NASA	National Aeronautics and Space Administration
NSIDC	National Snow and Ice Data Center (DAAC)
ODL	Object Definition Language
PCF	Process Control File
PDF	Portable Document Format
PDPS	Planning and Data Processing Subsystem
PGE	Product Generation Executive
PK	Primary Key
QA	Quality Assurance
SDSRV	Science Data Server CSCI
SQL	Structured Query Language
STMGT	Storage Management Software CSCI
SUBSRV	Subscription Server
WWW	World-Wide Web